PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-212071
(43)Date of publication of application : 20.08.1996

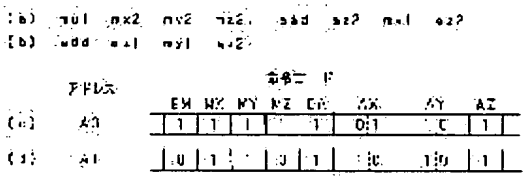
(51)Int.Cl. G06F 9/38

(21)Application number : 07-016189
(22)Date of filing : 02.02.1995

(71)Applicant : MITSUBISHI ELECTRIC CORP
(72)Inventor : ASAI TAKASHI
KAWAMOTO KOJI

(54) INSTRUCTION CODE GENERATION METHOD

(57)Abstract:
PURPOSE: To provide an instruction code generation method which sets the value preceding by one cycle of a computing element in an input designation field of the computing element where the arithmetic result of one cycle is not stored.
CONSTITUTION: An instruction code (c) is generated based on a mnemonic code (a) and stored in an address A0. The code (a) designates to store the result of multiplication of both contents mx2 and my2 of an input register as the contents mz2 of an output register and also to store the result of addition of both contents az2 and my1 of the input register as the contents az2 of the output register. Then an instruction code (d) is generated in a cycle following that where the code (c) is carried out and stored in an address A1. The code (d) instructs an adder to store the result of addition of contents mx1 and my1 of both registers as the contents az2 of the output register and not to store the result of multiplication of contents mx2 and my2 of both input registers as the contents mz1 of the output register based on a mnemonic code (b) which designates to store the result of addition of contents mx1 and my1 of the input register as the contents az2 of the output register.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision

of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号

特開平8－212071

(43)公開日　平成８年(1996)８月20日

(51)Int.Cl.[6]	識別記号	庁内整理番号	Ｆ Ｉ	技術表示箇所
Ｇ０６Ｆ　9/38	３７０　Ｘ			

審査請求　未請求　請求項の数４　ＯＬ　（全 ８ 頁）

(21)出願番号　特願平7－16189

(22)出願日　平成７年(1995)２月２日

(71)出願人　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目２番３号

(72)発明者　浅井　敬
兵庫県伊丹市瑞原４丁目１番地　三菱電機株式会社システムエル・エス・アイ開発研究所内

(72)発明者　川本　浩司
兵庫県伊丹市瑞原４丁目１番地　三菱電機株式会社システムエル・エス・アイ開発研究所内

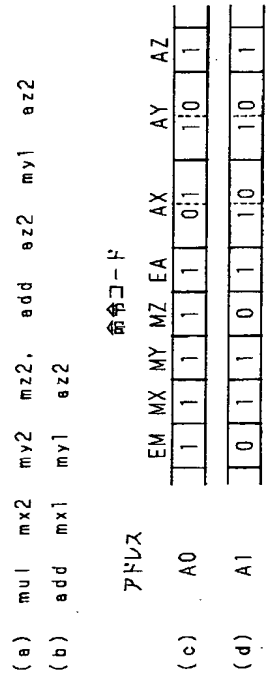
(74)代理人　弁理士　河野　登夫

(54)【発明の名称】　命令コード生成方法

(57)【要約】

【目的】　１のサイクルにおける演算結果を格納しない演算器の入力指定フィールドに、その演算器の１サイクル前の値を設定する命令コード生成方法の提供。

【構成】　入力レジスタの内容 mx2,my2を乗算した結果を出力レジスタの内容 mz2として格納し、入力レジスタの内容 az2,my1を加算した結果を出力レジスタの内容 az2として格納することを指定するニーモニックコード(a) に基づいて生成し、アドレスA0に格納してある命令コード(c) を実行した次のサイクルにおいて、入力レジスタの内容 mx1,my1を加算した結果を出力レジスタの内容az2 として格納することを指定するニーモニックコード(b) に基づき、加算器は両レジスタの内容mx1,my1 を加算した結果を出力レジスタの内容 az2として格納し、両入力レジスタの内容mx2,my2 を乗算した結果を出力レジスタの内容 mz1として格納しない命令コード(d) を生成し、アドレスA1に格納する。



1

【特許請求の範囲】

【請求項１】　プロセッサの各単位サイクル毎に演算を行う演算器に対して、前記演算器の演算の結果をレジスタに格納しないことを指定するフィールドと、前記演算器に入力すべきデータを指定する入力指定フィールドを含む命令コードを生成する命令コード生成方法において、

前記入力指定フィールドに、生成すべき命令コードが実行される単位サイクルの１サイクル前に実行すべく生成した前サイクル命令コードの前記演算器に対応した入力指定フィールドの値を設定することを特徴とする命令コード生成方法。

【請求項２】　前サイクル命令コードが１つである場合、該前サイクル命令コードの演算器に対応する入力指定フィールドの値を所定値とする請求項１記載の命令コード生成方法。

【請求項３】　前サイクル命令コードが複数ある場合、該複数の前サイクル命令コードのいずれか１つに含まれ演算器に対応する入力指定フィールドの値を所定値とする請求項１記載の命令コード生成方法。

【請求項４】　前サイクル命令コードが複数ある場合、該複数の前サイクル命令コードのうち実行回数が最も多い命令コードに含まれ演算器に対応する入力指定フィールドの値を所定値とする請求項１記載の命令コード生成方法。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は命令コード生成方法に関し、特に演算器の演算結果をレジスタに格納させない指定を含む命令コードを生成する命令コード生成方法に関する。

【０００２】

【従来の技術】図４は演算器を備えた本発明が対象とするマイクロプロセッサの１例を示すブロック図である。図において11,12 はデータを転送するバスであって、バス11はデータをMX_1 レジスタ13又はMX_2 レジスタ14へ与え、バス12はデータをMY_1 レジスタ15又はMY_2 レジスタ16へ与える。$MSEL_1$ セレクタ17は、両レジスタ13,14 のうちのいずれか一つを選択し、選択したレジスタの内容をMUL 乗算器19へ与える。$MSEL_2$ セレクタ18は、両レジスタ15,16 のうちのいずれか一つを選択し、選択したレジスタの内容をMUL 乗算器19へ与える。MUL 乗算器19は両セレクタ17,18から与えられる値を各サイクル毎に乗算し、乗算した結果をMZ_1 レジスタ20又はMZ_2 レジスタ21へ与える。

【０００３】$ASEL_1$ セレクタ22は、両レジスタ13,14 並びにAZ_1 レジスタ25及びAZ_2 レジスタ26のうちのいずれか一つを選択し、選択したレジスタの内容をADD 加算器24へ与える。$ASEL_2$ セレクタ23は、両レジスタ20,21 及び両レジスタ15,16 のうちのいずれか一つを選択し、選択したレジスタの内容をADD 加算器24へ与える。ADD加算器24は、両セレクタ22,23 から与えられた値を各サイクル毎に加算し、加算した結果を両レジスタ25,26 のいずれか一つへ与える。

2

【０００４】図５は図４のマイクロプロセッサを制御する命令コードの形式を説明する説明図であって、MUL 乗算器19及びADD 加算器24の制御に関するものを示す。図においてMXは$MSEL_1$ セレクタ17が選択するレジスタを指定する１ビットのフィールドである。MXの値が“０”の場合にMX_1 レジスタ13が指定され、MXの値が“１”の場合にMX_2 レジスタ14が指定される。MYは$MSEL_2$ セレクタ18が選択するレジスタを指定する１ビットのフィールドである。MYの値が“０”の場合にMY_1 レジスタ15が指定され、MYの値が“１”の場合にMY_2 レジスタ16が指定される。MZはMUL乗算器19が乗算した結果を格納するレジスタを指定する１ビットのフィールドである。MZの値が“０”の場合にMZ_1 レジスタ20が指定され、MZの値が“１”の場合にMZ_2 レジスタ21が指定される。EMはMUL 乗算器19が乗算した結果を格納するか否かを指定する１ビットのフィールドである。EMの値が“０”の場合にMUL 乗算器19の乗算結果はフィールドMZの指定に関係なく両レジスタ20,21 のいずれにも格納されない。EMの値が“１”の場合にMUL 乗算器19の乗算結果はフィールドMZで指定されたレジスタに格納される。即ちEMは乗算結果の格納の可否を指定するものであって、乗算の実行・非乗算を指定するものではない。

【０００５】AXは$ASEL_1$ セレクタ22が選択するレジスタを指定する２ビットのフィールドである。AXの値が“００”の場合にAZ_1 レジスタ25が指定され“01”の場合にAZ_2 レジスタ26が指定され、“10”の場合にMX_1 レジスタ13が指定され、“11”の場合にMX_2 レジスタ14が指定される。AYは$ASEL_2$ セレクタ23が選択するレジスタを指定する２ビットのフィールドである。AYの値が“00”の場合にMZ_1 レジスタ20が指定され、“01”の場合にMZ_2 レジスタ21が指定され、“10”の場合にMY_1 レジスタ15が指定され、“11”の場合にMY_2 レジスタ16が指定される。

【０００６】AZはADD 加算器24が加算した結果を格納するレジスタを指定する１ビットのフィールドである。AZの値が“０”の場合にAZ_1 レジスタ25が指定され、AZの値が“１”の場合にAZ_2 レジスタ26が指定される。EAはADD 加算器24が加算した結果を格納するか否かを指定する１ビットのフィールドである。EAの値が“０”の場合にADD 加算器24の加算結果はフィールドAZの指定に関係なく両レジスタ25,26のいずれかも格納されない。EAの値が“１”の場合にADD 加算器24の加算結果はフィールドAZで指定されたレジスタに格納される。即ちEAは加算結果の格納の可否を指定するものであって、加算の実行・非実行を指定するものではない。

【０００７】図６は図４のマイクロプロセッサを制御す

る命令コードで実行する操作を指定するニーモニックコードを説明する説明図である。図においてmul はMUL 乗算器19の乗算を制御するニーモニックコードであり、その後に両セレクタ17,18 が夫々選択するレジスタを指定するニーモニックコードが２つ及び乗算結果を格納するレジスタを指定するニーモニックコードが１つ随従する。このmul が記述されている場合、MUL 乗算器19の乗算結果は両レジスタ20,21 のいずれかに格納されるが、mul が記述されていない場合、両レジスタ20,21 のいずれにも格納されない。

【０００８】add はADD 加算器24の加算を制御するニーモニックコードであり、その後に両セレクタ22,23 が夫々選択するレジスタを指定するニーモニックコードが２つ及び加算結果を格納するレジスタを指定するニーモニックコードが１つ随従する。このadd が記述されている場合、ADD 加算器24の加算結果は両レジスタ25,26 のいずれかに格納されるが、add が記述されていない場合、両レジスタ25,26 のいずれにも格納されない。mx1,mx2,my1,my2,mz1,mz2,az1,az2 は、夫々MX_1 レジスタ13，MX_2 レジスタ14，MY_1 レジスタ15，MY_2 レジスタ16，MZ_1 レジスタ20，MZ_2 レジスタ21，AZ_1 レジスタ25，AZ_2 レジスタ26を指定するニーモニックコードである。そして１行に記述されたニーモニックコードは１サイクルで実行する命令を指定する。例えば、１行にニーモニックコードmul とadd とが記述されている場合、２つの演算を同じサイクルで実行させることを意味する。

【０００９】図６(a) はMX_2 レジスタ14の内容とMY_2 レジスタ16の内容とを乗算し、その乗算結果をMZ_2 レジスタ21に格納する操作とAZ_2 レジスタ26の内容とMY_1 レジスタ15の内容とを加算し、その加算結果をAZ_2 レジスタ26に格納する操作とを同時に行なうように指定するニーモニックコードである。図６(b) はMX_1 レジスタ13の内容とMY_2 レジスタ16の内容とを乗算し、その乗算結果をMZ_1 レジスタ20に格納する操作を指定するニーモニックコードである。図６(c) はMX_1 レジスタ13の内容とMY_1 レジスタ15の内容とを加算し、その加算結果をAZ_2 レジスタ26に格納する操作を指定するニーモニックコードである。マイクロプロセッサを制御する命令コードは図６で説明したニーモニックコードによる指定に従って生成される。

【００１０】図７は図６に示すニーモニックコードの指定に従い、従来の命令コード生成方法で生成された命令コードの説明図である。図において(f) は命令コードを構成する各フィールドの記号を示し、(a) は図６(a) のニーモニックコードに対して生成された命令コードを示し、(b) は図６(b) のニーモニックコードに対して生成された命令コードを示し、(c) は図６(c) のニーモニックコードに対して生成された命令コードを示す。

【００１１】従来の命令コード生成方法では、ニーモニックコードで指定されているフィールドにはその指定に

対応した値を設定し、ニーモニックコードで指定されていないフィールドには一定の値（以下デフォルトと称する）を設定する。あるサイクルにおいて乗算（又は加算）のみを行ない加算（又は乗算）を行わない場合、３個のフィールドAX,AY,AZ（又はMX,MY,MZ）にはデフォルトが設定される。従ってその場合両セレクタ17,18（又は22,23)は夫々特定のレジスタを選択する。

【００１２】図６(a) のニーモニックコードはMX_2 レジスタ14の内容とMY_2 レジスタ16の内容とを乗算し、乗算した結果をMZ_2 レジスタ21に格納し、同じサイクルでAZ_2 レジスタ26の内容とMY_1 レジスタ15の内容とを加算し、加算した結果をAZレジスタ26に格納すべきことを指定している故、生成した命令コードのEMフィールド,MXフィールド,MY フィールド,MZ フィールド,EA フィールド,AZ フィールドはいずれも“１”であり、AXフィールドは“01”であり、AYフィールドは“10”である。

【００１３】図６(b) のニーモニックコードはMX_1 レジスタ13の内容とMY_2 レジスタ16の内容とを乗算し、乗算した結果をMZ_1 レジスタ21に格納すべきことを指定している故、生成した命令コードのEMフィールドは“１”、MXフィールドは“０”、MYフィールドは“１”、MZフィールドは“０”である。そして乗算のみを行ない加算を行わない故、加算結果は両レジスタ25,26 のいずれにも格納しない。従って、EAフィールドは“０”である。デフォルトが“０”である場合、２桁の“０”は“00”となる故、AXフィールド，AYフィールドはいずれも“00”である。

【００１４】図６(c) のニーモニックコードはMX_1 レジスタ13の内容とMY_1 レジスタ15の内容とを加算し、加算した結果をAZ_2 レジスタ26に格納すべきことを指定している故、生成した命令コードのEAフィールドは“１”、AXフィールドは”10”、AYフィールドは“10”、AZフィールドは“１”である。そして加算のみを行ない乗算を行わない故、乗算結果は両レジスタ20,21 のいずれにも格納しない。従ってEMフィールドは“０”となる。デフォルトが“０”である場合、MXフィールド，MYフィールド，MZフィールドはいずれも“０”である。

【００１５】

【発明が解決しようとする課題】必要としない演算結果は、いずれのレジスタにも格納されない故、必要としない演算を行なう演算器へ与えるデータはどのような値であっても構わなかった。それ故、デフォルトを設定することにより命令コードの生成を簡略化することができた。しかしながら、このような命令コードの生成方法は、前記演算器で消費される消費電力を低減するように考慮されたものではなかった。

【００１６】前回のサイクルにおいて図７(a) に示される命令コードが実行され、今回のサイクルにおいて図７(b) に示される命令コードが実行される場合、MUL 乗算器19には前回のサイクルにおいて両レジスタ14,16 の内

容が入力され、今回のサイクルにおいて両レジスタ13,16の内容が入力される。$MSEL_1$ セレクタ17は前回のサイクルにおいてMX_2 レジスタ14を選択し、今回のサイクルにおいてMX_1 レジスタ13を選択している。異なるレジスタの内容が同一である割合は、そのビット長と相関がある。例えば、レジスタのビット長が16ビットである場合、レジスタに格納されている値がランダムな値である場合、異なるレジスタの内容が同一である場合は１／65536 である。

【００１７】$MSEL_2$ セレクタ18は、前回及び今回の連続した両サイクルにおいてMY_2 レジスタ16を選択している。同一のレジスタの内容が両サイクルの期間において一定である割合は、その期間にレジスタに対して書き込みが行われる頻度と相関がある。例えば、１サイクルの間に４個のレジスタの中からランダムに選ばれた１個のレジスタにデータを書き込む場合、同一のレジスタの内容が両サイクルの期間において一定である割合は３／４である。

【００１８】このように、前回及び今回の連続した両サイクルにおいて選択される両レジスタのうち一方が同じレジスタであり、他方が異なるレジスタである。そしてそれに伴いMUL 乗算器19に入力されるデータが一定である割合は低くなる。従ってMUL 乗算器19で消費される電力の増加が見込まれる。しかしEMフィールドで乗算結果の格納が指定されているので、消費電力の増加は演算に伴う必然のものであり問題にはならない。同様にしてADD 加算器24で消費される電力のより一層の増加が見込まれる。これはEAフィールドで加算結果の格納が指定されていないので不必要な電力消費であり問題である。この問題に対しAXフィールドに“01”を設定し、AYフィールドに“10”を設定するとよい。

【００１９】また、前回のサイクルにおいて図７(a) に示される命令コードが実行され、今回のサイクルにおいて図７(c) に示される命令コードが実行される場合、MUL 乗算器19には、前回のサイクルにおいて両レジスタ14,16 の内容が入力され、今回のサイクルにおいて両レジスタ13,15 の内容が入力される。即ち前回及び今回の連続した両サイクルにおいて選択される両レジスタはいずれも異なるレジスタである。そして、それに伴いMUL 乗算器19に入力されるデータが一定である割合は、より一層低くなる。従ってMUL 乗算19で消費される電力のより一層の増加が見込まれる。そしてEMフィールドで乗算結果の格納が指定されていないので不必要な電力消費であり、問題である。この問題に対し、MXフィールド及びMYフィールドにいずれも“１”を設定するとよい。同様にしてADD 加算器24で消費される電力の増加が見込まれる。これはEAフィールドで加算結果の格納が指定されているので消費電力の増加は問題にならない。

【００２０】このように今回のサイクルにおいて演算結果の格納が指定されていない演算器がある場合、その演算器にデータを入力すべきレジスタとして前回のサイクルにおいて選択したレジスタと同じレジスタを今回のサイクルにおいても選択する命令コードを生成することにより消費電力を節減できる命令コード生成方法を提供することを目的とする。

【００２１】

【課題を解決するための手段】第１発明に係る命令コード生成方法は、プロセッサの各単位サイクル毎に演算を行う演算器に対して、前記演算器の演算の結果をレジスタに格納しないことを指定するフィールドと、前記演算器に入力すべきデータを指定する入力指定フィールドを含む命令コードを生成する命令コード生成方法において、前記入力指定フィールドに、生成すべき命令コードが実行される単位サイクルの１サイクル前に実行すべく生成した前サイクル命令コードの前記演算器に対応した入力指定フィールドの値を設定することを特徴とする。

【００２２】第２発明に係る命令コード生成方法は、前サイクル命令コードが１つである場合、該前サイクル命令コードの演算器に対応する入力指定フィールドの値を所定値とすることを特徴とする。

【００２３】第３発明に係る命令コード生成方法は、前サイクル命令コードが複数ある場合、該複数の前サイクル命令コードのいずれか１つに含まれ演算器に対応する入力指定フィールドの値を所定値とすることを特徴とする。

【００２４】第４発明に係る命令コード生成方法は、前サイクル命令コードが複数ある場合、該複数の前サイクル命令コードのうち実行回数が最も多い命令コードに含まれ演算器に対応する入力指定フィールドの値を所定値とすることを特徴とする。

【００２５】

【作用】第１発明では、演算器に入力すべきデータを指定する入力指定フィールドに、生成すべき命令コードが実行される単位サイクルの１サイクル前に実行すべく生成した前サイクル命令コードの前記演算器に対応した入力指定フィールドの値を設定する。従って生成した命令コードを実行する単位サイクルとその１サイクル前とにおいて、演算器に入力されるデータが変化する割合が小さくなる。

【００２６】第２発明では、第１発明において前サイクル命令コードが１つである場合該前サイクル命令コードの演算器に対応する入力指定フィールドの値を所定値とする。従って生成した命令コードを実行する単位サイクルとその１サイクル前とにおいて、演算器に入力されるデータが変化する割合が小さくなる。

【００２７】第３発明では、第１発明において、前サイクル命令コードが２つ以上である場合、その中のいずれか１つに含まれ、演算器に対応する入力指定フィールドの値を所定値とする。従って生成した命令コードを実行する単位サイクルとその１サイクル前とにおいて、演算

器に入力されるデータが変化する割合が小さくなる。

【００２８】第４発明では、第１発明において、前サイクル命令コードが２つ以上ある場合、その中で実行回数が最も多い命令コードに含まれ、演算器に対応する入力指定フィールドの値を所定値とする。従って生成した命令コードを実行する単位サイクルとその１サイクル前とにおいて、演算器に入力されるデータが変化する割合が小さくなる。

【００２９】

【実施例】以下本発明をその実施例を示す図面に基づき具体的に説明する。本発明に係る命令コード生成方法により生成される命令コードで制御されるマイクロプロセッサは、図４に示されるものであって、乗算器及び加算器を備え、１サイクルにおいて乗算及び加算を必要とする場合又は乗算若しくは加算を必要とする場合に演算を行なう。そして後者の場合において両演算器は演算を行なうが、必要としない演算を行った演算器の演算結果は格納されない。生成される命令コードの形式は図５に示されるものであって、乗算結果を格納するか否かを指定する１ビットのフィールドがEMであり、乗算すべきデータを格納するレジスタを指定する１ビットのフィールドがMX及びMYであり、乗算結果を格納するレジスタを指定する１ビットのフィールドがMZである。

【００３０】また、加算結果を格納するか否かを指定する１ビットのフィールドがEAであり、加算すべきデータを格納するレジスタを指定する２ビットのフィールドがAX及びAYであり、加算結果を格納するレジスタを指定する１ビットのフィールドがAZである。生成すべき命令コードを指定するニーモニックコードは図６に示されるものであって、mul,add は夫々乗算，加算を制御するニーモニックコードであり、mx1,mx2,my1,my2,mz1,mz2,az1,az2 は夫々図４に示すレジスタMX_1，MX_2，MY_1，MY_2，MZ_1，MZ_2，AZ_1，AZ_2を示すニーモニックコードである。

【００３１】図１は第１実施例に係る命令コード生成方法を説明する説明図である。図において(a) はアドレスA0に配置される命令コードを指定するニーモニックコードを示し、(b) はアドレスA1に配置される命令コードを指定するニーモニックコードを示し、(c) はアドレスA0に配置された命令コードの一部を示し、既に生成されているものであり、(d) は生成してアドレスA1に配置すべき命令コードの一部を示す。アドレスA0に配置された命令コードが実行されたサイクルの次のサイクルにアドレスA1に配置された命令コードが実行される。

【００３２】図１(b) に示されるニーモニックコードはmul が記述されていない故、MUL 乗算器19の乗算結果はレジスタに格納されない。従って図１(d) に示す命令コードを生成する場合、その命令コードの１サイクル前で実行される命令コード、即ち図１(c) に示される命令コードを調べる。そして、図１(c) のMXフィールドの値１、MYフィールドの値１を図１(d) のMXフィールド，MYフィールドに設定する。その他のフィールドに対しては図１(b) のニーモニックコードで指定されているフィールドにはその指定に対応した値を設定し、ニーモニックコードで指定されていないフィールドにはデフォルトを設定する。即ち、MZフィールドにデフォルトとして“０”を、EAフィールドに加算結果の格納を示す“１”を、AXフィールドにMX_1 レジスタを示す“10”を、AYフィールドにMY_1 レジスタを示す“10”を、AZフィールドにAZ_2 レジスタを示す“１”を夫々設定し、アドレスA1に格納する。

【００３３】図１(d) に示される命令コードは、こうして生成された命令コードである。このようにすることにより、両アドレスA0,A1 に配置された命令コードを実行する２サイクルの期間においてMUL 乗算器19に入力させるべく選択されるレジスタは、MX_2 レジスタ14，MY_2 レジスタ16となる。即ち同一のレジスタの内容が入力される故、MUL 乗算器19で消費される電力は少なくなる。なお、本実施例においては、１サイクル前で実行される命令コードが１つ存在する場合について説明したが、１つも存在しない場合には図１(d) のMXフィールド，MYフィールドにデフォルトを設定する。また、割り込み処理ルーチンの先頭命令などのように、１サイクル前で実行される命令コードが予測できない場合は図１(d) のMXフィールド，MYフィールドにデフォルトを設定する。

【００３４】図２は第２実施例に係る命令コード生成方法を説明する説明図である。図において(a) はアドレスA0に配置される命令コードを指定するニーモニックコードを示し、(b) はアドレスA1に配置される命令コードを指定するニーモニックコードを示し、(c) はアドレスA2に配置される命令コードを指定するニーモニックコードを示す。(d) は既に生成されてアドレスA0に配置された命令コードの一部を示し、図示しない他のフィールドにはシーケンス制御命令が含まれ、そのシーケンス制御命令によりアドレスA0に配置された命令コードの次にアドレスA2に配置された命令コードが実行される。

【００３５】(e) は既に生成されてアドレスA1に配置された命令コードの一部を示し、図示しない他のフィールドにはシーケンス制御命令が含まれ、そのシーケンス制御命令によりアドレスA1に配置された命令コードの次にアドレスA2に配置された命令コードが実行される。(f) は生成してアドレスA2に配置すべき命令コードの一部を示す。

【００３６】図２(c) に示されるニーモニックコードはmul が記述されていない故、MUL 乗算器19の乗算結果はレジスタに格納されない。従って図２(f) に示す命令コードを生成する場合、その命令コードの１サイクル前で実行される命令コードを調べる。１サイクル前で実行される命令コードは図２(d) 及び図２(e) に示される２個の命令コードである。このように１サイクル前で実行さ

れる命令コードが複数個存在する場合、そのうちの任意の一つの命令コードのMXフィールド，MYフィールドの値を生成すべき命令コードのMXフィールド，MYフィールドに設定する。図２(f) の場合には図２(d) の命令コードの両フィールドMX,MY の値（１，１）を設定している。

【００３７】その他のフィールドに対しては図２(c) のニーモニックコードで指定されているフィールドには、その指定に対応した値を設定し、ニーモニックコードで指定されていないフィールドにはデフォルトを設定する。即ち、MZフィールドにはデフォルトとして“０”を、EAフィールドに加算結果の格納を示す“１”を、AXフィールドにAZ_2 レジスタを示す“01”を、AYフィールドにMY_1 レジスタを示す“10”を、AZフィールドにAZ_2 レジスタを示す“１”を夫々設定し、アドレスA2に格納する。

【００３８】このようにすることにより、両アドレスA0,A2 に配置された命令コードを実行する２サイクルの期間においてMUL 乗算器19に入力させるべく選択されるレジスタはMX_2 レジスタ14，MY_2 レジスタ16となる。また両アドレスA1,A2 に配置された命令コードを実行する期間においてMUL 乗算器19に入力させるべく選択されるレジスタは先のサイクルにおいてMX_1 レジスタ13，MY_2 レジスタ16となり、後のサイクルにおいてMX_2 レジスタ14，MY_2 レジスタ16となる。従来のデフォルトを設定する場合は、後のサイクルにおいてMX_1 レジスタ13，MY_1 レジスタ15が選択される故、本実施例の方が両サイクルにおいてレジスタの種類が変化する割合が小さく、MUL 乗算器19で消費される電力は少なくなる。即ち、乗算器の入力選択フィールドにデフォルトの値を設定する方式を用いた場合は、図６(d) の命令コードの後に図６(f) の命令コードが実行される時と図６(e) の命令コードの後に図６(f) の命令コードが実行される時のいずれの時も、MUL 乗算器19に入力されるレジスタの種類が変化する場合も生じるので、本発明の方式の方が入力されるレジスタの種類が変化する割合が少ない。

【００３９】図３は第３実施例に係る命令コード生成方法を説明する説明図である。図において(f) は生成してアドレスA2に配置すべき命令コードの一部を示す。(a)，(b) …(e) は図２と同様であるので説明を省略する。図３(f) に示す命令コードを生成する場合、その命令コードの１サイクル前で実行される命令コードを調べる。１サイクル前で実行される命令コードは図３(d) 及び図３(e) に示される２個の命令コードである。

【００４０】図３(d) に示される命令コードはN_a回実行され、図３(e) に示される命令コードはN_b回実行され、$N_a < N_b$である場合、図３(e) の命令コードのMXフィールド，MYフィールドの値（０，１）を生成すべき命令コードのMXフィールド,MYフィールドに設定する。図３(f) に示される命令コードは、こうして生成された命令コードである。このように１サイクル前で実行される命令コードが複数個存在する場合、その１サイクル前で実行される回数が最も多い命令コードの両フィールドMX,MY の値を生成すべき命令コードの両フィールドMX,MY に設定する。１サイクル前で実行される回数が最も多い命令コードが複数ある場合は、その中のいずれか一つの命令コードの両フィールドMX,MY の値を、生成すべき命令コードの両フィールドMX,MY に設定する。このようにすることにより、１サイクル前と現サイクルとにおいてMUL 乗算器19に入力させるべく選択されたレジスタの種類が変化する割合が小さく、MUL 乗算19で消費される電力は少なくなる。

【００４１】

【発明の効果】第１発明によれば、演算器に対応する生成すべき入力指定フィールドに、生成すべき命令コードが実行される単位サイクルの１サイクル前に実行すべく生成した前サイクル命令コードの前記演算器に対応した入力指定フィールドの値を設定する故、演算器に入力されるデータが変化する割合が小さくなり、演算器で消費される電力が少なくなる。

【００４２】第２発明によれば、第１発明において前サイクル命令コードが１つである場合、その前サイクル命令コードの演算器に対応する入力指定フィールドの値を所定値とする故、演算器に入力されるデータが変化する割合が小さくなり、演算器で消費される電力が少なくなる。

【００４３】第３発明によれは、第１発明において前サイクル命令コードが２つ以上である場合、その中のいずれか１つに含まれ、演算器に対応する入力指定フィールドの値を所定値とする故、演算器に入力されるデータが変化する割合が小さくなり、演算器で消費される電力が少なくなる。

【００４４】第４発明によれば、第１発明において前サイクル命令コードが２つ以上ある場合、その中で実行回数が最も多い命令コードに含まれ、演算器に対応する入力指定フィールドの値を所定値とする故、演算器に入力されるデータが変化する割合が小さくなり、演算器で消費される電力が少なくなる。

【図面の簡単な説明】

【図１】　第１実施例に係る命令コード生成方法を説明する説明図である。

【図２】　第２実施例に係る命令コード生成方法を説明する説明図である。

【図３】　第３実施例に係る命令コード生成方法を説明する説明図である。

【図４】　マイクロプロセッサの１例を示すブロック図である。

【図５】　図４に示すマイクロプロセッサを制御する命令コードの形式を説明する説明図である。

【図６】　図４に示すマイクロプロセッサを制御する命令コードを指定するニーモニックコードを説明する説明

11　　12

図である。

【図７】　従来の命令コード生成方法を説明する説明図である。

＊【符号の説明】

＊　13,14,15,16,20,21,25,26　レジスタ、17,18,22,23　セレクタ、19　乗算器、24　加算器。

【図１】

(a)　mul　mx2　my2　mz2,　add　az2　my1　az2

(b)　add　mx1　my1　az2

	アドレス	命令コード							
		EM	MX	MY	MZ	EA	AX	AY	AZ
(c)	A0	1	1	1	1	1	01	10	1
(d)	A1	0	1	1	0	1	10	10	1

【図２】

(a)　mul　mx2　my2　mz2,　add　az2　my1　az2

(b)　mul　mx1　my2　mz2,　add　az2　my1　az2

(c)　add　az2　my1　az2

	アドレス	命令コード							
		EM	MX	MY	MZ	EA	AX	AY	AZ
(d)	A0	1	1	1	1	1	01	10	1
(e)	A1	1	0	1	1	1	01	10	1
(f)	A2	0	1	1	0	1	01	10	1

【図３】

(a)　mul　mx2　my2　mz2,　add　az2　my1　az2

(b)　mul　mx1　my2　mz2,　add　az2　my1　az2

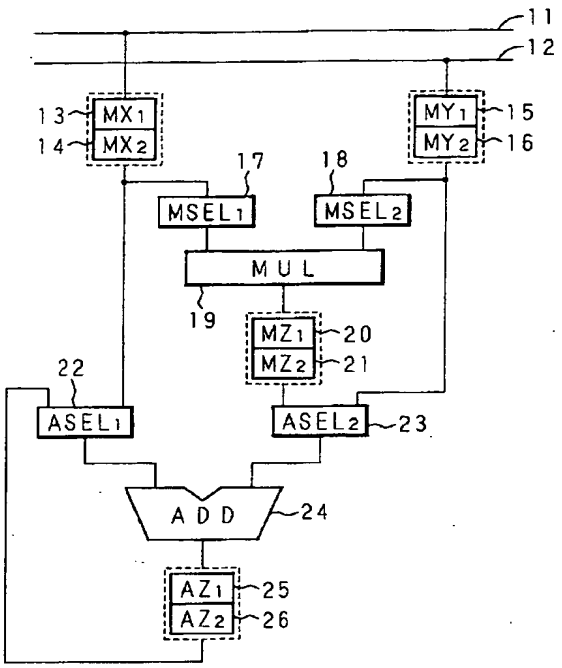
(c)　add　az2　my1　az2

	アドレス	命令コード							
		EM	MX	MY	MZ	EA	AX	AY	AZ
(d)	A0	1	1	1	1	1	01	10	1
(e)	A1	1	0	1	1	1	01	10	1
(f)	A2	0	0	1	0	1	01	10	1

【図５】

EM	MX	MY	MZ	EA	AX	AY	AZ

【図４】



11,12：バス
13,14,15,16,20,21,25,26：レジスタ
17,18,22,23：セレクタ
19：乗算器
24：加算器

【図７】

(f)	EM	MX	MY	MZ	EA	AX	AY	AZ
(a)	1	1	1	1	1	01	10	1
(b)	1	0	1	0	0	00	00	0
(c)	0	0	0	0	1	10	10	1

【図６】

```
(a)  mul  mx2  my2  mz2,  add  az2  my1  az2
(b)  mul  mx1  my2  mz1
(c)  add  mx1  my1  az2
```